Parameter Free Policy Shaping

***Abstract*—Policy Shaping is an algorithm that takes inputs as state-action-evaluation triples, that the learner obtains through interaction with a teacher. These triples are then used in combination with self exploration and a traditional reinforcement learning algorithm to learn a task. Policy Shaping has been experimentally shown to work well with noisy input from non-expert human teachers that are unfamiliar with the algorithm. Interactions with human teachers generate state-action-evaluation triples and, so far, the meaning of the evaluation part of the triple has always been hard coded into the algorithm. We present an algorithm that allows the learner to estimate the meaning of these labels automatically. The learner observes only unidentified evaluation labels, and then continuously re-estimates their meaning during learning and self exploration. Experiments with 30 human teachers, and several different types of simulated teachers, show that the algorithm is able to quickly understand the meaning of, and make use of: demonstrations, explicit action advice, and critique. For each of these three information sources, the parameter free algorithm strongly outperforms all static interpretations of labels when dealing with a set of teachers that have large internal variation in behavior. That is, when no single interpretation fits the full set of teachers that the learner interacts with, autonomously building an individual model for each teacher outperforms any a priori interpretation applied to the entire group.**

I. Introduction

Our work is aimed at allowing agents to learn from social interactions with human teachers. In particular, we are focused on learning from Critique, Demonstration, and Explicit Action Advice (EAA). During learning from critique, a human teacher observes an artificial learner take an action, and then has the opportunity to provide positive or negative critique. Demonstrations are generated by a human teacher taking successive actions. For Explicit Action Advice (EAA), a teacher is shown a state, and recommends an action. Critique is provided by a teacher observing state-action pairs in pac-man, demonstrations are provided by a teacher playing pac-man, and EAA is provided by a teacher observing states in pac-man. Each of these generate state-action-evaluation triples.

In Policy Shaping (9), a reinforcement learning algorithm builds a policy using self exploration, and this policy is continuously shaped based on the teacher's state-action-evaluation triples. Evaluations are interpreted as a noisy indicator of action choice quality and, depending on the type of evaluation received, increase or decrease the probability of taking an action. This interpretation has also been demonstrated to work well on critique from human teachers (10).

We describe and evaluate a Parameter Free version of Policy Shaping, that infers the meaning and reliability of social interactions. When interacting with a robot, it might be useful to be able to benefit from more than a set of pre specified keywords, allowing a teacher to interact with the learner using their own words, or even their own language. Instead of a pre specified keyword for "positive critique" a given teacher might prefer to use : "great", "ok", "excellent", "good robot", "duktig robot", "perfect", "yes", etc. In the case of critique, teachers were allowed to pick what keys on a keyboard to use for positive/negative critique on their own, and these had to be inferred by the algorithm. The meaning of demonstrated or recommended actions similarly needed to be inferred.

The algorithm was evaluated on data from 30 human teachers who provided demonstrations and critique on a pac-man board, as well as simulated teachers giving Explicit Action Advice (EAA). After gathering data, teachers were asked what the buttons that they used for critique meant. Results of the parameter free algorithm were compared with the results obtained when Policy Shaping was both given access to these answers, and also given the correct interpretation of demonstrated and recommended actions for demonstrations and EAA respectively.

The simulated teachers provided data with controllable noise levels. The meaning of labels was quickly learnt, even with noisy data, and the parameter free algorithm in general competed well with the case where the correct meaning of labels was hard coded. Using simulated teachers, we were able to vary teacher reliability in a systematic way, and create different sets of teachers. If the set of teachers was such that there existed a single interpretation that worked well for all of them, then the parameter free algorithm performed close to this universally correct interpretation. If the set of teachers was more varied, so that a single interpretation could not accurately describe all teachers in the set, the parameter free algorithm was able to significantly outperform all static models when learning simultaneously from the entire set of teachers.

II. Related work

Learning from interacting with humans has been explored in many different contexts. One common way of doing this is learning from demonstration (4). It has also been proposed that learning context-dependent skills could be achieved through inverse reinforcement learning; see (5) for early work and (6) for an overview. Instead of directly modelling the skill, a first inference step is performed, trying to infer the reward/cost function that the observed demonstrations are supposed to optimize. See (3) for a recent review of robots learning from human teachers.

In (1) it is suggested that humans might generate useful Explicit Action Advice (EAA), and that this advice could be used by a reinforcement learning algorithm in the same way as critique. As suggested, we use data from both EAA, as well as demonstrations, in the exact same way as we use critique; as

input to Policy Shaping. The interpretation of human critique as evaluations of an action choice is also explored in (2), that further shows that a learner can benefit from autonomously improving it's model of the teacher. The algorithm introduced in our paper shows that all constants of the Policy Shaping algorithm can be autonomously estimated by the learner for each of the three information sources investigated.

Combinations of evaluative feedback and demonstrations have been explored in (7), where the learner is provided with demonstrations, and the teacher is able to provide evaluative feedback by indicating parts of a reproduction where the learner performed either well or poorly.

Interpreting human behavior can be problematic, for example because humans make errors, and their behavior violates many assumptions of common machine learning algorithms in general, and reinforcement learning algorithms in particular (8; 11).

Human teachers will also, for example, use a mechanism meant as a channel for evaluating actions to try to motivate the learner. They might also try to evaluate actions that they think the learner might take in the future. In (13), a learner is described that takes advantage of feedback intended for future actions. One can also improve performance by including a button in the setup dedicated to motivational communication (11). The "motivate button" reduces the tendency of humans to use the evaluation channel for motivation purposes, and thus brings actual human behavior closer to the assumptions of the learning algorithm. To deal with certain types of flawed teachers, it is possible to explicitly model parts of the context that are visible to a human teacher and adapt the update mechanism based on this model (12).

Humans might also use positive and negative feedback in qualitatively different ways (15). Humans have a tendency to give more positive than negative reward, and to stop providing positive rewards when an agent appears to have learned the task, which can create problems if some common assumptions are made (8) (if a teacher always gives more positive than negative evaluations, and stops evaluating when "the robot is done learning", then slowing down learning can, for example, lead to increased total rewards).

The studies cited above motivate the removal of assumptions made about human teachers. The parameter free algorithm proposed in the presented paper makes no assumptions regarding the meaning of individual candidate action evaluations e. It instead checks if there is a correlation between a given teacher behavior/candidate action evaluation e and the quality of an action choice. If such a correlation is found for a given e, then it is used for updates (details below). (9) and (10) indicate that it is worth looking for this particular correlation, because they are often present in the signals of human teachers, and are useful during learning (if present and correctly interpreted).

Recently progress has started to be made with regards to automatically estimating how to interpret a human teacher's behavior, but this is still a fairly open field of research. In (17) an algorithm is presented that learns from a brain computer interface, autonomously estimating the meaning of the signals received during learning. An increasing number of experiments has started to show that it is possible to autonomously improve the interpretation of a teacher's signals (18; 14; 20; 19). The specific correlation type that the proposed parameter free algorithm checks for makes updates nice and straightforward. The fact that action choice quality is used, means that we have direct access to statistics on the exact values needed for the Policy Shaping update.

III. Reinforcement Learning

Typically, Reinforcement Learning (RL) defines a class of algorithms for solving problems modeled as a Markov Decision Process (MDP). An MDP is specified by the tuple (S, A, T, R, γ) for the set of possible world states S, the set of actions A, the transition function $T : S \times A \rightarrow P(S)$, the reward function $R : S \times A \rightarrow \mathbb{R}$, and a discount factor $0 \leq \gamma \leq 1$. We look for policies $\pi : S \times A \rightarrow \mathbb{R}$, mapping state-action pairs to probabilities, which result in high rewards. One way to solve this problem is through Q-learning (22). A Q-value $Q(s, a)$ is an estimate of the expected future discounted reward for taking action $a \in A$ in state $s \in S$. The Q-value of a state-action pair is updated based on the rewards received, and the resulting state. In this paper we use Boltzmann exploration (21) where the probability of taking an action is $Pr_q(a) = \frac{e^{Q(s,a)/\tau}}{\sum_{a'} e^{Q(s,a')/\tau}}$, where τ is a temperature constant.

Q-learning parameters were tuned without teacher data, and the values used were $T = 1.5$, $\alpha = 0.05$ and $\gamma = 0.9$.

IV. Policy Shaping

During critique, a teacher observes a pair consisting of an action a and a state s, and provides an action evaluation e, where state-action pairs are taken from some set of pairs. During EAA, a teacher observes a state s from some set of states, and provides a recommended action a. A "recommended" label is added to form a triple $(s, a, "recommended")$. During demonstrations, a teacher plays the game, showing the agent what it should do. A "demonstrated" label is added to each state action pair seen during playing (a demonstrator taking action a in state s leads to a $(s, a, "demonstrated")$). Data points are always in the form of a triplet of state s, action a and evaluation e. We only evaluate the algorithm on three information sources, but it can take inputs from any type of human behavior that is representable in such a (s, a, e) triplet. The meaning and reliability of a label can be either hard coded, or estimated.

A. The action choice quality interpretation of human generated critique

The model of human psychology that evaluations are of action choice quality is very straightforward: good actions lead to good evaluations, while bad actions lead to bad evaluations. By contrast, the assumptions needed to justify treating human feedback as a value to be maximized is contradicted by experimental evidence, and requires that the non-expert human maintain a rather complex model of learning: instead of

evaluating the action choice made by a learner, the teacher would need to keep track of and estimate the entire sequence of future rewards, and give a numerical estimate of this sum (so that, for example, an action creating a problem receives more negative reward than the sum of subsequent good action choices limiting the damage).

Finally, let's compare the two models in two different cases. First in the case where where only the final state matters, and secondly in the case where the specific actions matter and the final state is always the same, such as in dancing. If success can be measured in the final state, then the human teacher would need to make sure the total reward is path independent, making it necessary for the teacher to keep track of an enormous amount of information. However, if the goal is to perform the correct actions, i.e. correct dance moves, then the Policy Shaping interpretation: "evaluations refer to action choice" is favored almost by definition.

B. Advice with evaluations of known meaning

We have triples (s, a, e) of state s, action a and an action evaluation e of known meaning. In order to update the probability that an action a is correct in the state s, conditioned on the observation that (s, a) generated the action evaluation e, we need two probabilities: (i) $p(e|a = correct)$, the probability that label e will be observed conditioned on a being a correct action, and (ii) $p(e|a = incorrect)$, the probability that label e will be observed conditioned on a being an incorrect action.

For a prior probability p_{prior} that action a is correct in state s, we get:

$p_{posterior} = \frac{p(e|a=correct)p_{prior}}{p_{prior}p(e|a=correct)+(1-p_{prior})p(e|a=incorrect)}$

We make the approximation of conditional independence between individual (s, a, e) triples, allowing us to do one independent update per triplet.

If a specific state action pair was met with e_1 twice , and e_2 three times, we simply do the e_1 update 2 times and the e_2 update 3 times (where the posterior is used as the prior for the next update).

Starting with a uniform prior, and doing one update for every action evaluation instance gives a distribution. We also make the approximation of conditional independence between information sources, which allows us to get the final distribution by simply multiplying this distribution with the Boltzmann q-learning distribution. A specific meaning, or interpretation, is denoted $(p(e|a = correct), p(e|a = incorrect))$. If a label e has interpretation (0.6, 0.4), this means that it is more likely to show up as a response if the action was good than if it was bad. (0.8, 0.2) means the same thing, but with higher reliability.

C. Parameter free advice

For each type of action evaluation e encountered, the algorithm must now estimate the correlation between e and the quality of the action choice preceding the signal. This estimate then allows the learner to benefit from all instances of the feedback using Policy Shaping.

The two unknown probabilities are: $p(e|a = correct)$ and $p(e|a = incorrect)$. To estimate $p(e|a = correct)$ and $p(e|a = incorrect)$ for a given action evaluation e, we would ideally like to know the action choice quality of taking a in s for all triples (s, a, e) that contain e. Knowing what action should be taken, what action was actually taken, and what the label e was for all triples containing e would allow us to straightforwardly estimate the probability of e occurring as a response to good action choices, and as a response to bad action choices. The estimate of the meaning of e is however created during learning, before we know what to do everywhere (which is when we need the social feedback). For some of the triples containing e, we will have an estimate of the action choice quality, and for other triples we will have no idea if the action that generated the evaluation was good or not. For some of these triples, we will be more certain than for other triples, and we want to give extra weight to those while estimating the meaning of e. In other words, if an evaluation was seen after two different actions, and we know for certain that one of them was a good action, but think that the other was probably a bad action, then we would like the final interpretation to lean towards the evaluation corresponding to good actions.

We update the interpretation of each evaluation e after each game played, based on our current best guess of how good the action choices were that lead to e.

The Boltzmann Q-learning algorithm takes (s, a, r) triples, consisting of a state s, an action a, and a reward r. It also outputs a distribution of probabilities for taking an action π_Q. At each game, the agent obtains a new set of q-values from it's self exploration, and an associated policy π_Q. This policy can be interpreted as an estimate of how likely it is that each state action pair is a good action choice.

If we have multiple information sources, or are learning from multiple different teachers, we can however do better than π_Q. We are trying to learn the meaning of some label e_i. We now update π_Q on all e_k such that $k \neq i$ as described above, using our current best interpretation of each e_k, to get π_i. The probability P_C that an action a is a correct action choice in state s, is now taken to be $\pi_i(s, a)$ (our current best guess, based on current q-values, and current interpretation of all available labels). We will also need $P_I = 1 - P_C$ (the probability that the action is an Incorrect action choice). The two values we need to make the Policy Shaping update is $p(e|a = correct)$ and $p(e|a = incorrect)$. The procedure for obtaining them are detailed in algorithm 1, where the data set D is the set of all (s, a, e) triples.

V. Experimental Setup

Critique and demonstration data was gathered from 30 human teachers evaluating state-action pairs in the arcade game pac-man, and playing the game in a total of 7 sessions. Different types of simulated teachers provided critique, demonstrations and EAA.

procedure ALGORITHM 1(P_C, P_I, dataset D)
  $S_{correct} \leftarrow \emptyset$
  $S_{incorrect} \leftarrow \emptyset$
  for each $d_i \in D$ **do**
    $s_i, a_i, e_i \leftarrow state, action, evaluation \in d_i$
    if $P_C[s_i, a_i] > P_C[s_i, a_k] \forall a, k \neq i$ **then**
      $probability \leftarrow P_C[s_i, a_i]$
      $add(e_i, probability) to S_{correct}$
    end if
    if $P_I[s_i, a_i] > P_I[s_i, a_k] \forall a, k \neq i$ **then**
      $probability \leftarrow P_I[s_i, a_i]$
      $add(e_i, probability) to S_{incorrect}$
    end if
  end for
  $S_C = \sum probability \in S_{correct}$
  $S_I = \sum probability \in S_{incorrect}$
  for each type of evaluation e_i **do**
    $p(e_i|a = correct) \leftarrow$
  $(\sum probability \in S_{correct}$, where evaluation = $e_i)/S_C$
    $p(e_i|a = incorrect) \leftarrow$
  $(\sum probability \in S_{incorrect}$, where evaluation = $e_i)/S_I$
  end for
  return $p(e|a = correct)$, $p(e|a = incorrect)$
end procedure

Fig. 1. Parameter Free Policy Shaping

A. Domain

We use the experimental domain of pac-man, because human teachers are easily familiar with it. Pac-man consists of a 2-D grid with food, walls, ghosts, and the pac-man avatar. Eating all food pellets ends the game with +500 reward, and being killed by the ghost ends the game with -500 reward. Each food pellet gives +10 reward, and each time step pac-man gets a -1 time penalty. Pac-man's action space is to go up, down, right or left. The state representation includes pac-man's position, the position and orientation of any ghosts and the presence of food pellets. In this version of pac-man, the ghost moves in a random direction each time step, but does not go back to a square it has just occupied.

B. User study

We solicited participation from the campus community and had 30 volunteers provide data for this experiment. Participants provided demonstrations and evaluated videos in 7 different sessions. Each session contained 300 time steps, generated 300 data points, and lasted between 2 and 4 minutes, depending on how much time the teacher had to choose actions or give critique. In figure 2 we see the starting position of the pac-man board used in this experiment. Some sessions primary function was to familiarize users with the setup. The setup was also designed to generate data for future studies on an algorithm that autonomously learns the meaning of labels. There was also always a tradeoff between gather as much data as possible, and not boring the users (bored or annoyed users might generate strange data).

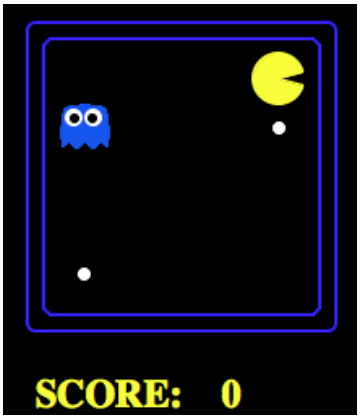

Fig. 2. During demonstrations, the teachers played the game with the starting position shown above. During critique, they provided feedback on videos consisting of multiple games, where each game had the same starting position.

The seven sessions were:

- 1: Positive Demonstrations: providing positive actions (playing the game to win), with severe time constraints.
- 2: Critique of an agent taking random actions, with severe time constraints.
- 3: Positive Demonstrations: providing positive actions.
- 4: Negative Demonstrations: providing negative actions (playing to lose; showing what not to do).
- 5: Critique of an agent taking random actions.
- 6: Critique of an agent taking good actions.
- 7: Critique of an agent taking bad actions.

The total setup took just under 30 minutes of pressing buttons in front of the screen, with a total of around 40 minutes including instructions and questions. Between each session the teacher had a short break, receiving instructions about the next session, and/or answering questions about the previous session or the setup in general. This was done in order to learn lessons for future setup designs, to generate data for possible future work looking at correlations between answers and performance, and to give participants a chance to rest.

To generate the three different sets of state-action pairs for critique, simulated agents played the game for 300 time steps, one random agent, and two based on the policy π_{sim}. To create π_{sim}, Boltzmann q-learning was run from 10 different different starting states to better cover the state space, creating 10 sets of q-values (for each starting state, the algorithm was run for a long time until q-values stagnated). π_{sim} was then generated by taking the sum of the 10 q-values. The sessions with good/bad actions took the most/least probable action of π_{sim} with probability 0.8, and otherwise an action was chosen randomly amongst the other actions. The resulting behavior was clearly biased towards good/bad actions, but without monotonically always wining/loosing.

C. Experimental setup

We generated 8 experimental conditions that we tested our algorithm in. The first one was the case with no teacher, simply referred to as "no teacher". The second was with human

Critique and Demonstrations, as well as noise free simulated EAA, referred to as "C+D+EAA".

Then we added 6 different types of data sets with noisy simulated teachers. Each data set contained a number of different teachers, all of the same information source. For each information source we generated sets consisting of 3 teachers with noise levels: $n = 0$, $n = 0.2$ and $n = 0.4$ respectively (denoted "C with 3 teachers" for critique, "D with 3 teachers" for demonstrations, and "EAA with 3 teachers" for EAA).

For each information source we also generated sets consisting of 6 teachers with noise levels: $n = 0$, $n = 0.2$, $n = 0.4$, $n = 0.6$, $n = 0.8$, and $n = 1.0$ respectively (denoted "C with 6 teachers" for critique, etc).

Each dataset consists of a set of state, action, evaluation triples which are generated by a simulated teachers that produces data with noise n. To create simulated teachers we started with π_{sim} mentioned above.

For critique of a state action pair (s, a) such that the a has the highest probability according to π_{sim} in s, the a simulated teacher with noise n generates an evaluation called "positive critique" with a probability $(1 - n)$. Otherwise it generates an evaluation called "negative critique". For critique of a state action pair (s, a) such that the a does not have the highest probability according to π_{sim} in s, a simulated teacher with noise n generates an evaluation called "negative critique" with a probability $(1 - n)$. Otherwise it generates an evaluation called "positive critique".

For the EAA the simulated teacher recommended an action a for each given state s. We used the same states that were used in session 5 of the user study (states visited by an agent taking random actions). For a given state s, the action a with the highest value in π_{sim} was recommended with probability of $(1-n)$, and another random action otherwise. This produced the triple (s, a "recommended").

In the case of demonstrations, the simulated played the game, following the policy π_{sim} with probability $(1 - n)$.

VI. Results

A single run consist of running 500 learning episodes and storing the score at the end of each episode. We used the average score of all 500 episodes as a measurement of the performance of the algorithm on some specific data set. We ran 600 such runs for each result reported below, giving us fairly narrow 99 percent confidence intervals (each simulated teacher generated 600 separate data sets, and the learning algorithm was run 20 times on the data for each of the 30 human teachers). Each number reported in table 1 is thus an average of 600 independent runs (and 300 000 games).

As we can see in figure 3, the parameter free algorithm quickly learns the meaning of the labels. We can also see that in this case, the various static interpretations tested have identical (or at least very nearly identical) performance, showing the robustness of the algorithm.

In table 1 we can see that when we have three teachers, the parameter free algorithms performance is similar to the various static assumptions (where each static assumption correspond to attaching an estimated reliability to an accurate prior knowledge of the meaning of each label).

We can also see that in all three cases where we have 6 teachers, the parameter free algorithm strongly outperforms all static assumptions. This is probably due to the fact that these 6 teachers are so different that no single interpretation is able to accurately model all teachers.

We also tried to see what happens if we give incorrect interpretations to labels for the various data sets. For all data sets, and for all reliability estimates, this led to performance worse than the case with no data. In all tested conditions, the agent did however eventually learn to play the game and reliably win (even when reality and estimate were diametrically opposite and fixed, the agent were eventually able to learn, even tough it took a bit longer than the case with no teacher).

TABLE I
Data set comparison

Data	Interpretation	Average Score	σ
No Teacher	n/a	342.3 ± 2.31	21.97
C+D+EAA	parameter free	482.1 ± 1.70	16.18
C+D+EAA	**(0.6,0.4)**	485.4 ± 1.63	15.49
C+D+EAA	(0.7,0.3)	484.9 ± 1.63	15.50
C+D+EAA	(0.8,0.2)	483.4 ± 1.57	14.93
C with 3 teachers	**parameter free**	484.2 ± 1.35	12.81
C with 3 teachers	(0.6,0.4)	479.3 ± 1.54	14.62
C with 3 teachers	(0.7,0.3)	478.6 ± 1.55	14.76
C with 3 teachers	(0.8,0.2)	480.9 ± 1.41	13.43
C with 6 teachers	**parameter free**	490.0 ± 1.13	10.74
C with 6 teachers	(0.6,0.4)	360.5 ± 3.37	32.00
C with 6 teachers	(0.7,0.3)	347.6 ± 3.65	34.70
C with 6 teachers	(0.8,0.2)	396.3 ± 2.36	22.45
D with 3 teachers	parameter free	491.4 ± 1.39	13.19
D with 3 teachers	**(0.6,0.4)**	497.2 ± 1.07	10.18
D with 3 teachers	(0.7,0.3)	484.6 ± 1.44	13.67
D with 3 teachers	(0.8,0.2)	479.7 ± 1.54	14.64
D with 6 teachers	**parameter free**	491.7 ± 1.41	13.38
D with 6 teachers	(0.6,0.4)	471.6 ± 1.58	15.04
D with 6 teachers	(0.7,0.3)	474.7 ± 1.64	15.64
D with 6 teachers	(0.8,0.2)	471.3 ± 1.76	16.70
EAA with 3 teachers	parameter free	437.3 ± 2.35	22.38
EAA with 3 teachers	**(0.6,0.4)**	443.9 ± 1.98	18.79
EAA with 3 teachers	(0.7,0.3)	440.1 ± 2.05	19.52
EAA with 3 teachers	(0.8,0.2)	438.6 ± 1.94	18.44
EAA with 6 teachers	**parameter free**	458.8 ± 1.71	16.30
EAA with 6 teachers	(0.6,0.4)	395.5 ± 2.40	22.87
EAA with 6 teachers	(0.7,0.3)	395.0 ± 2.31	22.00
EAA with 6 teachers	(0.8,0.2)	438.6 ± 1.94	18.44

VII. Discussion

We have shown that the presented algorithm can significantly outperform any static interpretation if it is interacting with a group of teachers that is so diverse that no single model works well for all member of the group. We have also shown that when the group of teachers are similar enough for a single model to describe them all, the presented algorithm can quickly find a model that performs at, or very near to, the case where this model is given a priori. These findings were reproduced in several different types of social interactions, with similar results seen for critique, demonstrations, or explicit action advice.

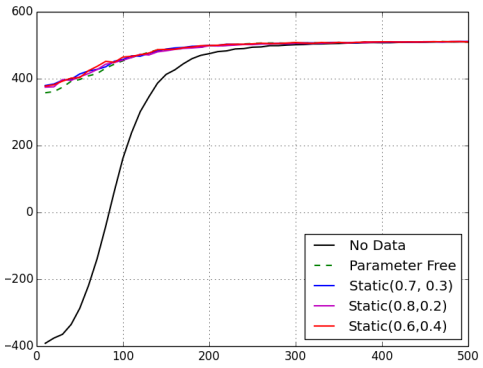

Fig. 3. C+D+EAA. This shows us the results of learning simultaneously from human generated Critique, Demonstrations, and simulated Explicit Action Advice. The, much lower, black line shows the case of no teacher, and the other lines show learning with Policy Shaping, either using a fixed interpretation, or the parameter free algorithm. We can see that the parameter free algorithm (the green line) is slightly below the other lines in the beginning, but that it quickly catches up. See table I for the average score over the entire learning interval of 500 games, as well as confidence intervals.

REFERENCES

[1] Gabriel V. de la Cruz Jr., Bei Peng, Walter S. Lasecki, and Matthew E. Taylor. Generating real-time crowd advice to improve reinforcement learning agents. In Learning for General Competency in Video Games workshop (AAAI), January 2015.

[2] R. Loftin, J. MacGlashan, B. Peng, M. E. Taylor, M. L. Littman, J. Huang, and D. L. Roberts. A Strategy-Aware Technique for Learning Behaviors from Discrete Human Feedback. *Proceedings of AAAI*.2014.

[3] Sonia Chernova and Andrea L. Thomaz. Robot learning from human teachers. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2014.

[4] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot Programming by Demonstration. *In Siciliano, B. and Khatib, O. (eds.). Handbook of Robotics*, pp. 1371-1394. Springer. 2008.

[5] A.Y. Ng, and S. Russell. Algorithms for inverse reinforcement learning, *Proceedings of the Seventeenth International Conference on Machine Learning*. pp 663-670. 2000.

[6] G. Neu, and C. Szepesvári. Training parsers by inverse reinforcement learning. *Machine learning*, vol 77, number 2, pp 303-337. 2009.

[7] B.D. Argall, B. Browning, and M. Veloso. Teacher feedback to scaffold and refine demonstrated motion primitives on a mobile robot. *Robotics and Autonomous Systems*. 59(3-4). pp 243-255. 2011.

[8] C. L. Isbell, M. Kearns, S. Singh, C. Shelton, P. Stone, D. Kormann. Cobot in LambdaMOO: An Adaptive Social Statistics Agent. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2006.

[9] S. Griffith, K. Subramanian, J. Scholz C.L. Isbell, and A. L. Thomaz. Policy Shaping: Integrating Human Feedback with Reinforcement Learning. *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, 2013.

[10] T. Cederborg, I. Grover, C. L. Isbell, and A. L. Thomaz. Policy Shaping With Human Teachers. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[11] A. L. Thomaz and C. Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners *Artificial Intelligence Journal*, 172:716-737, 2008.

[12] C. Breazeal, J. Gray, and M. Berlin. An Embodied Cognition Approach to Mindreading Skills for Socially Intelligent Robots, *I. J. Robotic Res.* vol. 28, 5, pp 656-680. 2009.

[13] Knox, W.B., Breazeal, C., Stone, P.: Learning from feedback on actions past and intended. *Proceedings of 7th ACM/IEEE International Conference on Human-Robot Interaction HRI*. 2012.

[14] T. Cederborg and P-Y. Oudeyer. A Social Learning Formalism for Learners Trying to Figure Out What a Teacher Wants Them to Do *PALADYN Journal of Behavioral Robotics*, 2014, 5, pp 64-99.

[15] A. L. Thomaz and C. Breazeal. Asymmetric Interpretations of Positive and Negative Human Feedback for a Social Learning Agent, *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2007.

[16] A. L. Thomaz and C. Breazeal. Experiments in Socially Guided Exploration: Lessons learned in building robots that learn with and without human teachers. *Connection Science, Special Issue on Social Learning in Embodied Agents*, pages 91-110, 2008.

[17] Grizou, Jonathan and Iturrate, Iñaki and Montesano, Luis and Oudeyer, Pierre-Yves and Lopes, Manuel. Calibration-Free BCI Based Control *AAAI Conference on Artificial Intelligence*. 2014.

[18] M. Lopes, T. Cederborg, P-Y. Oudeyer. Simultaneous acquisition of task and feedback models. *IEEE International Conference on Development and Learning (ICDL)*. 2011.

[19] I. Iturrate, J. Grizou, J. Omedes, P-Y. Oudeyer, M. Lopes and L. Montesano. Exploiting task constraints for self-calibrated brain-machine interface control using error-related potentials. *Plos One*. 2015.

[20] R. Loftin, B. Peng, J. MacGlashan, M. L. Littman, M. E. Taylor, J. Huang, and D. L. Roberts. Learning Something from Nothing: Leveraging Implicit Human Feedback Strategies. *Proceedings of the Twenty-Third IEEE International Symposium on Robot and Human Communication (ROMAN)*. 2014.

[21] C. J. Watkins. Models of Delayed Reinforcement Learning. PhD thesis, Psychology Department, Cambridge University, 1989.

[22] C. Watkins and P. Dayan, "Q learning: Technical note," Machine Learning, vol. 8, no. 3-4, pp. 279-292, 1992.